APMA 2580
Multiscale Computational Fluid Dynamics
Instructor: Professor George Karniadakis
Location: 170 Hope Street, Room 108
Time: Tuesday 10:30 - 1:00 pm

Today’s topic:

**Bottom-Up Models for CFD**
Part I: Molecular Dynamics (MD)
Part II: Dissipative Particle Dynamics (DPD)

By Zhen Li
Email: zhen_li@brown.edu
May 3, 2016
Part I: Molecular Dynamics

• Basics of MD simulation
  • motivation, history, typical length and time scales

• Potentials
  • Non-bonded interactions
  • Bonded interactions

• Algorithms for time integration
  • Verlet, velocity-Verlet, Beeman

• How do you practically run a MD simulation
  • Scaling, Periodic BC, Potential cut-offs, cell-list and Verlet list, thermostats

• Analysis of MD
  • Configurations, time correlations, transport properties
Continuum mechanics vs. atomistic viewpoint

**Continuum assumption: (PDEs)**
- Material can be modeled as a continuous mass that fills the entire region of space it occupies.
- No underlying inhomogeneous microstructure, that is, matter can be divided infinitely without change of material properties.
- It ignores the fact that matter is made of atoms.
- Only valid on length scales much greater than that of inter-atomic distances.

**Atomistic viewpoint: (Newton’s second law $F=ma$)**
- Material is made of discrete atoms.
- No spatial discretization necessary – given by atomic distances
- More fundamental description of the world, does not distinguish different subjects (physics, chemistry, biology, material science, et. al.)
Length and time scales for different models

- **Atomistic Models**
- **Mesoscopic Models**
- **Continuum Equations**

**Time**:
- femtoseconds (fs) $10^{-15}$
- picoseconds (ps) $10^{-12}$
- nanoseconds (ns) $10^{-9}$
- microseconds (μs) $10^{-6}$
- milliseconds (ms) $10^{-3}$
- seconds (s) $10^{-1}$
- minutes (min) $10^{0}$

**Length**:
- nanometers (nm) $10^{-9}$
- micrometers (μm) $10^{-6}$
- millimeters (mm) $10^{-3}$
- meters (m) $10^{0}$

Basics of MD simulation

If all of scientific knowledge were to be destroyed, and only one sentence passed on to the next generations of creatures, what statement would contain the most information in the fewest words?

**Richard Feynman:** “I believe it is the *atomic hypothesis* (or the atomic *fact*, or whatever you wish to call it) that *all things are made of atoms.*”

**Motivation of MD simulation**

The computer experiments (*in-silico* experiments):

- For a better understanding of underlying mechanisms of real experiments

  ![Image of snowflake](image1)

  **2002,** Molecular dynamics simulation of the ice nucleation and growth process leading to water freezing, M Matsumoto, S Saito and I Ohmine
  *Nature* **416**, 409-413

- Allow to study the dynamic processes at atomistic level
  - Dynamical events control processes which affect functional properties of the biomolecules.
  - Fast dynamic processes of crack development

![Image of molecular structure](image2)
Basics of MD simulation

History

Rahman published a landmark simulation, establishing the field.

Correlations in the Motion of Atoms in Liquid Argon*

A. Rahman
Argonne National Laboratory, Argonne, Illinois
(Received 6 May 1964)

A system of 864 particles interacting with a Lennard-Jones potential and obeying classical equations of motion has been studied on a digital computer (CDC 3600) to simulate molecular dynamics in liquid argon at 94.4 K and a density of 1.374 g cm⁻³. The pair-correlation function and the constant of self-diffusion are found to agree well with experiment; the latter is 15% lower than the experimental value. The spectrum of the velocity autocorrelation function shows a broad maximum in the frequency region \( \omega = 0.25 (k_B T / \hbar) \). The shape of the Van Hove function \( G_0(r,t) \) attains a maximum departure from a Gaussian at about \( t = 3.0 \times 10^{-12} \) sec and becomes a Gaussian again at about \( 10^{-11} \) sec. The Van Hove function \( G_0(r,t) \) has been compared with the convolution approximation of Vineyard, showing that this approximation gives a too rapid decay of \( G_0(r,t) \) with time. A delayed-convolution approximation has been suggested which gives a better fit with \( G_0(r,t) \); this delayed convolution makes \( G_0(r,t) \) decay as \( t^4 \) at short times and as \( t \) at long times.

Only 864 atoms,
System properties, structure of atoms and coefficient of self-diffusion, compared well with experimental data
Basics of MD simulation

History

1975, Computer simulation of protein folding
M Levitt, A Warshel
*Nature* 253:94.
(~750 atoms)

1979, Dynamics of ligand binding to heme protein
DA Case, M Karplus
*J Mol Biol* 132:343

M. Levitt, A. Warshel together with M. Karplus received the 2013 Nobel Prize in Chemistry awarded in part for the application of MD to proteins.

2015, Nothing to Sneeze At: A Dynamic and Integrative Computational Model of an Influenza A Virion
*Structure* 23, 584–597
(> 10 M particles)
Basics of MD simulation

Typical length and time scales of MD systems

Length scale:
Limited by the number of atoms that can be included in the simulation.
Typical MD systems contain thousands to millions of atoms.
System size: several nanometres to hundreds of nanometres.

\[(1 \, nm = 10^{-9} m)\]

Time scale:
Time step is limited by the smallest oscillation period of fastest atomic motions.
Simulation times: picoseconds to nanoseconds.

\[(1 \, ps = 10^{-12} s, 1 \, ns = 10^{-9} s)\]

\[f \sim 100THz = 10^{14}s^{-1}\]
A MD system is made of many atoms

- \( N \) particles
- Mass \( m_i \)
- Position \( r_i \)
- Velocity \( v_i \)

**Total energy of system**

\[
E = K + V = \frac{1}{2} \sum_{i=1}^{N} m_i v_i^2 + \sum_{i=1}^{N} V(r_i)
\]

**Equation of motion for atoms**

\[
m_i \frac{d^2 \mathbf{r}_i}{dt^2} = -\nabla_{r_i} V(\mathbf{r}_i)
\]

- System of coupled 2\(^{nd}\) order nonlinear differential equations
- Solved by discretizing in time (spatial discretization given by individual atoms)
Basics of MD simulation

Procedure of MD simulation

Initial atomic model

- \( r_i(t = 0) \)
- \( v_i(t = 0) \)

Calculate forces acting on each atom

- \( F = -\nabla V(r_i) \)

Move each atom according to those forces

- \( \frac{d r_i}{d t} = v_i \)
- \( m \frac{d v_i}{d t} = F_i \)

Advance simulation time by a time step
Part I: Molecular Dynamics

• Basics of MD simulation
  • motivation, history, typical length and time scales

• Potentials
  • Non-bonded interactions
  • Bonded interactions

• Algorithms for time integration
  • Verlet, velocity-Verlet, Beeman

• How do you practically run a MD simulation
  • Scaling, Periodic BC, Potential cut-offs, cell-list and Verlet list, thermostats

• Analysis of MD
  • Configurations, time correlations, transport properties
Non-Bonded interactions

**Van der Waals interaction** is referred to as the combination of attractive and repulsive forces between two atoms, which are not bonded to each other.

Lennard-Jones potential:

\[
V_{LJ} = 4\varepsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right]
\]

The attractive part (power 6) has been experimentally validated. The repulsive part (power 12) is empirical.

Lorentz-Berthelot mixing rules: \( \sigma_{ij} = \frac{1}{2}(\sigma_{ii} + \sigma_{jj}) \), \( \varepsilon_{ij} = \sqrt{\varepsilon_{ii} \varepsilon_{jj}} \)

**Coulomb interaction** is the electrostatic forces between two atoms.

\[
V_E = \frac{q_i q_j}{4\pi \varepsilon_{ij} r_{ij}}
\]
Vibration of O-H bond
Bond length: $r_0 = 0.958\text{Å}$
Frequency: $f = 101.9$ THz $\approx 10^{14} \text{s}^{-1}$
Period: $T \approx 10^{-14} \text{s}$
($\text{Å} = 10^{-10} \text{m}, \text{nm} = 10^{-9} \text{m}$)
($fs = 10^{-15} \text{s}, ps = 10^{-12} \text{s}, ns = 10^{-9} \text{s}$)
Bonded interactions 2: Bond angle bending

Vibration of O-H angle bending
Bond length: $\theta_0 = 104.45^\circ$
Frequency: $f = 47.8 \text{ THz} \approx 5 \times 10^{13} \text{s}^{-1}$
Period: $T \approx 2 \times 10^{-14} \text{s}$
Potentials

Bonded interactions 3: Proper dihedral angle bending (Torsion)

\[ V_{\text{torsion}} = K_\phi [1 + \cos(n\phi_{ijkl})] \]

Torsion:
Periodicity needs to be enforced.
Potentials

Bonded interactions 4: Improper dihedral angle bending

\[ V_{dihedral} = K_\omega (\omega - \omega_0)^2 \]
Potentials

\[ V(r_i \ldots r_N) = \sum_{ij} 4\epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \]

\[ + \sum_{ij} \frac{q_i q_j}{4\pi \epsilon_{ij} r_{ij}} \]

\[ + \sum_{\text{bond}} K_b (r_{ij} - r_0)^2 \]

\[ + \sum_{\text{angle}} K_\theta (\theta_{ij} - \theta_0)^2 \]

\[ + \sum_{\text{torsion}} K_\phi [1 + \cos(n\phi_{ijkl})] \]

\[ + \sum_{\text{dihedral}} K_\omega (\omega_{ij} - \omega_0)^2 \]
Part I: Molecular Dynamics

• Basics of MD simulation
  • motivation, history, typical length and time scales

• Potentials
  • Non-bonded interactions
  • Bonded interactions

• Algorithms for time integration
  • Verlet, velocity-Verlet, Beeman

• How do you practically run a MD simulation
  • Scaling, Periodic BC, Potential cut-offs, cell-list and Verlet list, thermostats

• Analysis of MD
  • Configurations, time correlations, transport properties
Integrator: Verlet Algorithm

Taylor expansion for particle position at $t + \Delta t$:

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{1}{2}\Delta t^2 a(t) + O(\Delta t^3)$$

Similarly, the old position at $t - \Delta t$:

$$r(t - \Delta t) = r(t) - v(t)\Delta t + \frac{1}{2}\Delta t^2 a(t) - O(\Delta t^3)$$

Combine the above two equation, we have

$$r(t + \Delta t) = 2r(t) - r(t - \Delta t) + \Delta t^2 a(t) + O(\Delta t^3)$$

Thus the velocity at $t$ is:

$$v(t) = \dot{r}(t) = \frac{1}{2\Delta t} (r(t + \Delta t) - r(t - \Delta t)) + O(\Delta t^2)$$

- Velocities not explicitly solved.
- A two-step method
- Advantages: simplicity and good stability
- Global error $O(\Delta t^2)$
**Integrator: velocity-Verlet Algorithm**

Taylor expansion for particle position and velocity at $t + \Delta t$:

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{1}{2} \Delta t^2 a(t) + O(\Delta t^3)$$

$$v(t + \Delta t) = v(t) + a(t)\Delta t + \frac{1}{2} \dot{a}(t)\Delta t^2 + O(\Delta t^3)$$

Taylor expand acceleration

$$\dot{a}(t)\Delta t^2 = \left(\frac{a(t + \Delta t) - a(t)}{\Delta t}\right)\Delta t^2 + O(\Delta t^3)$$

Thus the position and velocity are updated by:

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{1}{2} a(t)\Delta t^2 + O(\Delta t^3)$$

$$v(t + \Delta t) = v(t) + \frac{1}{2} \Delta t(a(t + \Delta t) + a(t) + O(\Delta t^3)$$

- Velocity calculated explicitly
- Possible to control the temperature
- Stable in long time simulation
- Most commonly used algorithm
Algorithms for time integration

Each integration cycle using velocity-Verlet Algorithm

\[
\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{1}{2}(\Delta t)^2 \mathbf{f}_i(t),
\]
\[
\mathbf{\tilde{v}}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{1}{2} \Delta t \mathbf{f}_i(t),
\]
\[
\mathbf{f}_i(t + \Delta t) = \mathbf{f}_i(\mathbf{r}(t + \Delta t), \mathbf{\tilde{v}}(t + \Delta t)),
\]
\[
\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{1}{2} \Delta t (\mathbf{f}_i(t) + \mathbf{f}_i(t + \Delta t)).
\]

Beeman Algorithm

\[
r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{1}{6} \left( 4a(t) - a(t - \Delta t) \right) \Delta t^2 + O(\Delta t^3)
\]
\[
v(t + \Delta t) = v(t) + \frac{1}{6} \left( 2a(t + \Delta t) + 5a(t) - a(t - \Delta t) \right) \Delta t + O(\Delta t^3)
\]
Part I: Molecular Dynamics

• Basics of MD simulation
  • motivation, history, typical length and time scales

• Potentials
  • Non-bonded interactions
  • Bonded interactions

• Algorithms for time integration
  • Verlet, velocity-Verlet, Beeman

• How do you practically run a MD simulation
  • Scaling, Periodic BC, Potential cut-offs, cell-list and Verlet list, thermostats

• Analysis of MD
  • Configurations, time correlations, transport properties
How do you practically run a MD simulation?

1. Parameterization of MD system
   - Scaling by model parameters
     - Length scale: size of atom $\sigma$
     - Energy scale: depth of LJ potential well $\varepsilon$
     - Mass scale: mass of the atom $m$
   - Other units can be determined by $\sigma$, $\varepsilon$ and $m$
     - Time unit: $\tau = \sigma (m/\varepsilon)^{1/2}$
     - Velocity unit: $v = (m/\varepsilon)^{1/2}$
     - Force unit: $F = \varepsilon/\sigma$
     - Pressure unit: $P = \varepsilon/\sigma^3$
     - Temperature unit: $T = \varepsilon/k_B$

An example: liquid argon at 120K:
We choose units of mass, length and energy, then the reduced LJ units become $\sigma = 1$, $m = 1$, $\varepsilon = 1$.

<table>
<thead>
<tr>
<th>Physical quantity</th>
<th>Unit</th>
<th>Value for Ar</th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
<td>$\sigma$</td>
<td>$3.4 \times 10^{-10}$ m</td>
</tr>
<tr>
<td>energy</td>
<td>$\varepsilon$</td>
<td>$1.65 \times 10^{-21}$ J</td>
</tr>
<tr>
<td>mass</td>
<td>$m$</td>
<td>$6.69 \times 10^{-26}$ kg</td>
</tr>
<tr>
<td>time</td>
<td>$\sigma(m/\varepsilon)^{1/2}$</td>
<td>$2.17 \times 10^{-12}$ s</td>
</tr>
<tr>
<td>velocity</td>
<td>$(\varepsilon/m)^{1/2}$</td>
<td>$1.57 \times 10^2$ m/s</td>
</tr>
<tr>
<td>force</td>
<td>$\varepsilon/\sigma$</td>
<td>$4.85 \times 10^{-12}$ N</td>
</tr>
<tr>
<td>pressure</td>
<td>$\varepsilon/\sigma^3$</td>
<td>$4.20 \times 10^7$ N·m$^{-2}$</td>
</tr>
<tr>
<td>temperature</td>
<td>$\varepsilon/k_B$</td>
<td>120 K</td>
</tr>
</tbody>
</table>
How do you practically run a MD simulation?

1. Parameterization of MD system

Lennard-Jones potential energy function

\[
U(r) = 4\varepsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right]
\]

\[
F(r) = -\frac{dU(r)}{dr} = \frac{24\varepsilon}{\sigma} \left[ \frac{2}{r} \left( \frac{\sigma}{r} \right)^{13} - \left( \frac{\sigma}{r} \right)^7 \right]
\]
How do you practically run a MD simulation?

2. Get the initial configuration

Simple systems: initial position of particles can be randomly initialized.
Complex systems: taken from data banks, for example Protein Data Bank (www.rcsb.org)

3. Assign initial velocities

At thermal equilibrium, the mean kinetic energy of the system at temperature $T$ is:

$$
\langle K \rangle = \frac{1}{2} \sum_{i=1}^{3N} m_i v_i^2 = \frac{1}{2} (3N) k_B T
$$

This can be obtained by assigning the velocity components $v_i$ from a random Gaussian distribution with zero mean and standard deviation $(k_B T/m_i)$. 
How do you practically run a MD simulation?

4. Apply periodic boundary conditions

For computational box $0 < x < L$

If $x < 0$ then $x = x + L$
If $x > L$ then $x = x - L$

Minimum image criterion: among all images of a particle, consider only the closest and neglect the rest.

5. Truncate non-bonded interactions

**Bonded interactions:** local, therefore the computational cost is $O(N)$.

**Non-bonded interactions:** involve all pairs of atoms, therefore the cost is $O(N^2)$.

Reducing the computing cost: use of cut-off for non-bonded interactions.
The cutoff distance may be no greater than $\frac{1}{2} L$ ($L$= box length)

1. Abrupt truncation
2. Switching
3. Shifting
How do you practically run a MD simulation?

6. Choosing the time step

- Too small: covering small conformation space
- Too large: numerical instability
- Suggested time steps
  - Translation, 10 fs
  - Flexible molecules and rigid bonds, 2 fs
  - Flexible molecules and bonds, 1 fs
7. Saving CPU time

**Cell list**

Update cell list at every time steps

$Cell_i = \left[ \frac{r_i}{r_c} \right]$  

**Verlet list**

Update verlet list only when

$|\Delta r|_{max} > r_v - r_c$
How do you practically run a MD simulation?

7. Saving CPU time

An example:

Computational Cost: $O(N^2)$

With Lists, computational cost: $O(N)$

Long-range electrostatic interactions $O(N^2)$:

- Ewald summation (Ewald, 1921): $\sim O(N^{3/2})$
- Fast multipole method (Greengard, 1987): $\sim O(N)$
- Particle mesh Ewald (Darden, 1993): $\sim O(N\log N)$
How do you practically run a MD simulation?

8. Numerical temperature control (thermostats)

- Basic MD
  - NVE (micro-canonical ensemble): an adiabatic process with no heat exchange. Total energy of the system is conserved.

- Other common ensembles
  - NVT (canonical ensemble): constant temperature MD. In NVT, the energy of endothermic and exothermic processes is exchanged with a thermostat.

  - NPT (isothermal-isobaric ensemble): In addition to a thermostat, a barostat is needed. It corresponds most closely to laboratory conditions with a flask open to ambient temperature and pressure.

  - μVT (grand-canonical ensemble): chemical potential, temperature and volume are constants. The system exchange energy and particles with a reservoir, so that various possible states of the system can differ in both their total energy and total number of particles.
How do you practically run a MD simulation?

8. Numerical temperature control (thermostats)

1. Andersen thermostat
   
   The simplest thermostat which does correctly sample the NVT ensemble.
   
   At each step, some prescribed number of particles is selected, and their velocities are drawn from a Gaussian distribution at the prescribed temperature.

   \[ P(v) = \left( \frac{\beta}{2\pi m} \right)^{3/2} \exp\left[ -\beta m v^2 / 2 \right] \]

   The strength of the coupling to the heat bath is specified by a collision frequency, \( \nu \). For each particle, a random variate is selected between 0 and 1. If this variate is less than \( \nu \Delta t \), then that particle's velocities are reset.

2. Langevin thermostat
   
   At each time step, all particles receive a friction force and a random force that satisfy the fluctuation-dissipation theorem, thereby guaranteeing NVT statistics.

   \[ m_i \frac{d^2 r_i}{dt^2} = -\nabla_i V - \gamma v_i + W_i(t) \]

   where \( \gamma \) is a friction coefficient, and \( W_i(t) \) is a random force with

   \[ \langle W_i(t)W_j(t') \rangle = \delta_{ij} \delta(t - t') k_B T \]
How do you practically run a MD simulation?

8. Numerical temperature control (thermostats)

3. Nose-Hoover thermostat

A Hamiltonian with an extra degree of freedom for heat bath, $s$, is introduced

$$H_{Nose} = \sum_{i=1}^{N} \frac{p_i^2}{2m_i s^2} + V(q_i) + \frac{p_s^2}{2Q} + gk_B T \ln(s)$$

According to the Hamiltonian formalism, we define the equations of motion by using the extended Hamiltonian

$$\begin{align*}
\frac{dq_i}{dt} &= \frac{\partial H_{Nose}}{\partial p_i} = \frac{p_i}{m_i s^2} \\
\frac{dp_i}{dt} &= -\frac{\partial H_{Nose}}{\partial q_i} = -\frac{\partial \phi}{\partial q_i} \\
\frac{ds}{dt} &= \frac{\partial H_{Nose}}{\partial p_s} = \frac{p_s}{Q} \\
\frac{dp_s}{dt} &= -\frac{\partial H_{Nose}}{\partial s} = \sum \frac{p_i^2}{m_i s^2} - gkT
\end{align*}$$

It can be approved that the partition function of the extended system is equivalent to that of the original system in the canonical ensemble except for a constant factor.

4. Dissipative particle dynamics thermostat (will be introduced in Part II.)
How do you practically run a MD simulation?

Procedure of MD simulation

1. **Initial atomic model**
   - $r_i(t = 0)$
   - $v_i(t = 0)$

2. **Calculate forces acting on each atom**
   - $F = -\nabla V(r_i)$

3. **Move each atom according to those forces**
   - $\frac{dr_i}{dt} = v_i$
   - $m \frac{dv_i}{dt} = F_i$

4. **Advance simulation time by a time step**
Part I: Molecular Dynamics

- **Basics of MD simulation**
  - motivation, history, typical length and time scales
- **Potentials**
  - Non-bonded interactions
  - Bonded interactions
- **Algorithms for time integration**
  - Verlet, velocity-Verlet, Beeman
- **How do you practically run a MD simulation**
  - Scaling, Periodic BC, Potential cut-offs, cell-list and Verlet list, thermostats
- **Analysis of MD**
  - Configurations, time correlations, transport properties
**Analysis of MD**

- **Static properties** such as structure, energy, and pressure are obtained from **pair (radial) distribution functions**

\[ g(r) = \frac{n(r)}{\frac{4\pi r^2 \Delta r}{\rho}} \]

- \( g(r) \, dr \) is the probability of finding a particle in volume \( d^3r \) around \( r \) given one at \( r=0 \)
- \( g(r) \rightarrow 0 \) as \( r \rightarrow 0 \) due to the strong repulsive forces
- \( g(r) \) tends to 1 as \( r \) at large distances, no long-range order
- \( g(r)=1 \) for Ideal gas
other outputs

1. Kinetic Energy

\[
K = \lim_{t \to \infty} \frac{1}{t} \int_{t_0}^{t_0+t} \sum_i \frac{1}{2} m_i v_i^2(\tau) \, d\tau
\]

According to the equipartition theorem, we have

\[
\left\langle \frac{1}{2} m_i v_i^2 \right\rangle = \frac{d}{2} N k_B T
\]

where \( d \) is the dimensionality.

This defines the temperature of MD system.

2. Potential Energy

\[
\bar{V} = \lim_{t \to \infty} \frac{1}{t} \int_{t_0}^{t_0+t} \sum_{i<j} V(\mathbf{r}_i(\tau) - \mathbf{r}_j(\tau)) \, d\tau = \frac{N \rho}{2} \int V(\mathbf{r}) g(\mathbf{r}) d^3r
\]

3. Pressure

\[
P = \rho k_B T + \frac{2\pi \rho^2}{3} \int_0^\infty dr r^3 f(r) g(r)\quad\quad P = \rho k_B T + \frac{1}{3V} \left( \sum_i \mathbf{r}_i \cdot \mathbf{F}_i \right)
\]
dynamic and transport properties are obtained from time correlation functions

Velocity autocorrelation function

\[ \psi(t_d) = \frac{1}{N} \left( \sum_{i} \overline{v_i}(t_0) \cdot \overline{v_i}(t_0 + t_d) \right) \]

Kinetic energy \[ \psi(0) = \frac{3kT}{m} \]

Diffusivity \[ D = \int_{0}^{\infty} \psi(t_d) \]

Stress autocorrelation function

\[ J = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} \]

\[ \varphi(t_d) = \frac{\rho}{3kT} \frac{1}{N} \sum_{\alpha\beta} \langle J_{\alpha\beta}(t_0) J_{\alpha\beta}(t_0 + t_d) \rangle \]

Shear modulus \[ \varphi(0) \]

Viscosity \[ \eta = \int_{0}^{\infty} \varphi(t_d) \]
Analysis of MD

transport properties

Example: 1D diffusion

\[ \frac{\partial N}{\partial t} = D \frac{\partial^2 N}{\partial x^2} \]

\[ N(x, t) = \frac{N_0}{2\sqrt{\pi Dt}} \exp \left[ -\frac{x^2}{4Dt} \right] \]

Second moment of the distribution is the mean-square displacement

\[ \langle [x(t) - x(0)]^2 \rangle = \frac{1}{N_0} \int x^2 N(x, t) dx \]

Einstein’s relation

\[ \langle [x(t) - x(0)]^2 \rangle = 2Dt \]

Applicable when the time is large compared to the average time between atomic collisions

At short time

\[ x(t) = vt \]

\[ [x(t)]^2 = [vt]^2 \sim t^2 \]
transport properties

Example: 1D diffusion

\[
\dot{x}(t) = \frac{dx}{dt} \quad x(t) - x(0) = \int_0^t \dot{x}(t') dt'
\]

Square both sides and average over time origins

\[
msd = \left\langle [x(t) - x(0)]^2 \right\rangle = \int_0^t dt'' \int_0^t dt' \left\langle \dot{x}(t') \dot{x}(t'') \right\rangle
\]

Use integrand symmetry, shift the time origin, then we obtain

\[
\left\langle [x(t) - x(0)]^2 \right\rangle = \frac{t}{2t} = \int_0^t d\tau \left\langle \dot{x}(\tau)\dot{x}(0) \right\rangle \left(1 - \frac{\tau}{t} \right)
\]

Take the long-time limit

\[
\lim_{t \to \infty} \left\langle [x(t) - x(0)]^2 \right\rangle = \int_0^\infty d\tau \langle \dot{x}(\tau)\dot{x}(0) \rangle
\]

Consider Einstein’s relation

\[
D = \int_0^\infty d\tau \langle \dot{x}(\tau)\dot{x}(0) \rangle = \frac{1}{3} \int_0^\infty \langle \mathbf{v}(t)\mathbf{v}(0) \rangle dt
\]

Green-Kubo relation (3D)
Examples of MD simulation of simple fluids


Precursor Film in Dynamic Wetting, Yuan and Zhao, PRL, 2010.

Effect of nano structures on the nucleus wetting modes, Xu et al., RSC Adv. 2016.
Software/Package for MD simulation

**LAMMPS**:  http://lammps.sandia.gov/

**GROMACS**:  http://www.gromacs.org/

**NAMD**:  http://www.ks.uiuc.edu/Research/namd/

**HOOMD-blue**:  http://codeblue.umich.edu/hoomd-blue/

AMBER
CHARMM
Materials Studio
RedMD
ESPResSo  ...  ...

Part II: Dissipative Particle Dynamics

- **Brief introduction of DPD**
  - motivation, history
- **Foundations of DPD**
  - Coarse-graining of MD (Mori-Zwanzig formalism)
- **Parameterization of a DPD system**
  - Force field of classic DPD
  - Pressure, compressibility, viscosity, diffusivity of DPD fluids
  - Scaling
  - Modeling of complex fluids using DPD
- **How do you practically run a DPD simulation**
  - Implementation of boundary conditions
  - Computing viscosity, diffusivity
- **Some Applications**
  - Droplet, Blood flows, Self-assembly Dynamics
Why CG/mesoscopic?

MACROscale
Continuum description

Navier-Stokes equation

MESOscale: A Scale Between Them

Discontinuum Nature
Thermal Fluctuation
Beyond the capacity of MD

~ 5.0 million per mm³

MICROscale
Atomistic description

Continuum fluid mechanics

Coarse-graining of molecules
Loses dynamical details of atom/molecules
Constitutes thermal fluctuation

Mesh: FDM, FEM, FVM ...
Mesh-free: SPH

Mesh: LBM, FH
Mesh-free: CGMD, DPD, SDPD

Molecular dynamics

Atomistic description
Contains details of atom/molecules
Limited in time/length

Mesh-free: MD
Length and time scales for different models

Dissipative Particle Dynamics (DPD)
Mesoscopic Models
Atomistic Models
Quantum Mechanics
Fluctuating hydrodynamics

History of DPD method

- **Babyhood (1992-1995)**
  - Original formulation (Hoogerbrugge & Koelman, 1992)

- **Youth (1995-2003)**
  - Fluctuation-dissipation relation (Español & Warren, 1995)
  - Important contributions to the DPD methodology (model of polymers, implementation of boundary conditions, et al.)

- **Golden Era (2003-now)**
  - Successful applications to material science, biological and biomedical systems, fluid rheology and other complex fluids.
Successful DPD applications
Governing equation of DPD

- Particles in DPD represent clusters of molecules and interact through simple pair-wise forces

\[ F_i = \sum_{i \neq j} \left( F^C_{ij} + F^D_{ij} + F^R_{ij} \right) \]

\[ F^C_{ij} = \alpha (1 - r_{ij}/r_c) e_{ij} \]
\[ F^D_{ij} = \gamma (1 - r_{ij}/r_c)^2 (e_{ij} v_{ij}) e_{ij} \]
\[ F^R_{ij} = \sigma (1 - r_{ij}/r_c) dt^{-1/2} \xi_{ij} e_{ij} \]

- DPD system is thermally equilibrated through a thermostat defined by forces \( F^D_{ij} \) and \( F^R_{ij} \) via \( \sigma^2 = 2\gamma k_B T \).

- The time evolution equations are given by:

\[ d\mathbf{r}_i = \mathbf{v}_i \, dt, \quad d\mathbf{v}_i = \mathbf{F}_i \, dt \]

Hoogerbrugge & Koelman, EPL., 1992
Part II: Dissipative Particle Dynamics

• Brief introduction of DPD
  • motivation, history

• Foundations of DPD
  • Coarse-graining of MD (Mori-Zwanzig formalism)

• Parameterization of a DPD system
  • Force field of classic DPD
  • RDF, pressure, compressibility, viscosity, diffusivity of DPD fluids
  • Scaling
  • Modeling of complex fluids using DPD

• How do you practically run a DPD simulation
  • Implementation of boundary conditions
  • Computing viscosity, diffusivity

• Some Applications
  • Droplet, Blood flows, Self-assembly Dynamics
Where does DPD come from?

DPD is a bottom-up particle-based **mesoscopic** method from coarse-graining of MD system.

**Microscopic system**
- All-atom model
- MD

**Mesoscopic system**
- Coarse-grained model
- DPD

Irrelevant variables are eliminated
Consider a linear differential system for two variables:
\[
\frac{dx}{dt} = x + y, \tag{1}
\]
\[
\frac{dy}{dt} = -y + x, \tag{2}
\]

Let \(x_0 = x(t = 0)\) and \(y_0 = y(t = 0)\) denote the corresponding initial values. By solving the Eq. (2)
\[
y = \int_0^t e^{-(t-s)}x(s)ds + y_0 e^{-t}
\]
we can reduce the system into an equation for \(x(t)\) alone:
\[
\frac{dx}{dt} = x + \int_0^t e^{-(t-s)}x(s)ds + y_0 e^{-t}
\]
The second term in above equation introduces memory.

**Dimension Reduction** leads to memory effect and noise term.
Mori-Zwanzig Formalism (Zwanzig, Nonequilibrium Statistical Mechanics, 2001)

\[
\frac{d\phi_i(x, t)}{dt} = R(\phi(x, t)), \quad \phi_i(x, 0) = x_i, \quad 1 \leq i \leq n.
\]

\[\phi = (\hat{\phi}, \tilde{\phi}), \quad \hat{\phi} = (\phi_1, \ldots, \phi_m), \quad \tilde{\phi} = (\phi_{m+1}, \ldots, \phi_n)\]

Form the Liouville equation \( u_t = Lu \), the components \( \hat{\phi} \) are

\[\hat{\phi}_j(x, t) = e^{tL}x_j\]

Let \( \mathbb{P} \) be the conditional expectation projection \( \mathbb{P}g(x) = E[g|\hat{x}] \).

Define \( \mathbb{Q} = I - \mathbb{P} \) and keep in mind that \( \mathbb{P}^2 = \mathbb{P}, \mathbb{Q}^2 = \mathbb{Q} \), and \( \mathbb{PQ} = 0 \), as must be true for any projection.

\[\frac{\partial}{\partial t} e^{tL}x_j = Le^{tL}x_j = e^{tL}Lx_j = e^{tL}\mathbb{P}Lx_j + e^{tL}\mathbb{Q}Lx_j\]

Dyson’s formula

\[e^{tL} = e^{tQL} + \int_0^t e^{(t-s)L}\mathbb{P}Le^{sQL}ds\]

\[\frac{\partial}{\partial t} e^{tL}x_j = e^{tL}\mathbb{P}Lx_j + \int_0^t e^{(t-s)L}\mathbb{P}Le^{sQL}QLx_j ds + e^{tQL}QLx_j\]

This is the Mori-Zwanzig equation. This equation is exact and is an alternative way of writing the original system.
Mori-Zwanzig Formalism (Zwanzig, Nonequilibrium Statistical Mechanics, 2001)

Consider an atomistic system consisting of $N$ atoms which are grouped into $K$ clusters, and $N_C$ atoms in each cluster.

The Hamiltonian of the atomistic system is:

$$H = \sum_{\mu=1}^{K} \sum_{i=1}^{N_C} \frac{p_{\mu,i}^2}{2m_{\mu,i}} + \frac{1}{2} \sum_{\mu,n} \sum_{i,j \neq i} V_{\mu,i,j}$$

Our interest is on the molecular or CG level:

The equation of motion for CG particles can be written as:

$$\dot{P}_{\mu} = k_B T \frac{\partial}{\partial R_{\mu}} \ln \omega(R)$$

$$- \frac{1}{k_B T} \sum_{X=1}^{K} \int_{0}^{t'} ds \langle [\delta F_{I}(t-s)] [\delta F_{X}(0)^T] \rangle \cdot \frac{P_{X}(s)}{M_{X}}$$

$$+ \delta F_{I}(t)$$

Details see Z. Li et al, Soft Matter, 10, 8659, 2014.
Bottom-up coarse-grained model:

The equation of motion (EOM) of the coarse-grained (CG) particles obtained from the Mori-Zwanzig projection is in a form of generalized Langevin equation, which is given by

\[
\frac{d}{dt} \mathbf{P}_I = \frac{1}{\beta} \frac{\partial}{\partial \mathbf{R}_I} \ln \omega(\mathbf{R}) - \beta \sum_{X=1}^K \int_0^t ds \left\langle [\delta \mathbf{F}_I(t-s)][\delta \mathbf{F}_X(0)]^T \right\rangle \frac{\mathbf{P}_X(s)}{M_X} + \delta \mathbf{F}_I(t),
\]

First approximation: Here, we assume that the non-bonded interactions between neighboring clusters in the microscopic system are explicitly pairwise decomposable, and hence the total force consists of pairwise forces, e.g. \( \mathbf{F}_I \approx \sum_{J \neq I} \mathbf{F}_{IJ} \) and \( \delta \mathbf{F}_I \approx \sum_{J \neq I} \delta \mathbf{F}_{IJ} \).

Second approximation: In practice, we neglect the many-body correlations between different pairs, and assume that the force \( \mathbf{F}_{IJ} \) between two clusters \( I \) and \( J \) depends only on the relative COM positions \( \mathbf{R}_I \) and \( \mathbf{R}_J \) and is independent of the positions of the rest of clusters.
Evaluation of coarse-grained interactions:

First term: Conservative Force:

\[
\frac{1}{\beta} \frac{\partial}{\partial \mathbf{R}_I} \ln \omega(\mathbf{R}) = \langle \mathbf{F}_I \rangle \approx \sum_{J \neq I} \langle \mathbf{F}_{IJ} \rangle = \sum_{J \neq I} F^C_{IJ}(R_{IJ}) e_{IJ}
\]

Second term: Dissipative Force:

\[
- \beta \sum_{X=1}^{K} \int_0^t ds \langle [\delta \mathbf{F}_I(t-s)][\delta \mathbf{F}_X(0)]^T \rangle \frac{P_X(s)}{M_X}
\]

Based on the second approximation, the correlation of fluctuating forces between different pairs is ignored. Thus, we have

\[
\langle [\delta \mathbf{F}_I(t-s)][\delta \mathbf{F}_X(0)]^T \rangle \frac{P_X(s)}{M_X}
\]

\[
= \sum_{J \neq I} \sum_{Y \neq X} \langle [\delta \mathbf{F}_{IJ}(t-s)][\delta \mathbf{F}_{XY}(0)]^T \rangle V_X(s)
\]

\[
= \sum_{J \neq I} \langle [\delta \mathbf{F}_{IJ}(t-s)][\delta \mathbf{F}_{IJ}(0)]^T \rangle V_J(s) |_{X=I, Y=J} + \sum_{J \neq I} \langle [\delta \mathbf{F}_{IJ}(t-s)][\delta \mathbf{F}_{JI}(0)]^T \rangle V_J(s) |_{X=J, Y=I}
\]

\[
= \sum_{J \neq I} \langle [\delta \mathbf{F}_{IJ}(t-s)][\delta \mathbf{F}_{IJ}(0)]^T \rangle V_{IJ}(s)
\]
Markovian approximation:

Remark: The memory term given by Eq. (8) can be further simplified with a Markovian assumption that the memory of fluctuating force in time is short enough to be approximated by a Dirac delta function

\[
\beta \langle [\delta F_{IJ}(t - s)] [\delta F_{IJ}(0)]^T \rangle = 2 \gamma_{IJ} \delta(t - s), \\
\beta \int_0^t ds \langle [\delta F_{IJ}(t - s)] [\delta F_{IJ}(0)]^T \rangle V_{IJ}(s) = \gamma_{IJ} \cdot V_{IJ}(t),
\]

where \( \gamma_{IJ} \) is the friction tensor defined by \( \gamma_{IJ} = \beta \int_0^\infty dt \langle [\delta F_{IJ}(t)] [\delta F_{IJ}(0)]^T \rangle \). Then, the equation of motion of DPD particles based on the Markovian approximation can be expressed by

\[
\frac{dP_I}{dt} = \sum_{J \neq I} \{ F_{IJ}^C(R_{IJ}) e_{IJ} - \gamma_{IJ}(R_{IJ}) (e_{IJ} \cdot V_{IJ}) e_{IJ} + \delta F_{IJ}(t) \}
\]

DPD model

Where does DPD come from?

Answer: DPD comes from coarse-graining of its underlying microscopic system.

- Irrelevant variables are eliminated using MZ projection.
- Only resolve the variables that we are interested in.
- Unresolved details are represented by the dissipative and random forces.
Part II: Dissipative Particle Dynamics

• Brief introduction of DPD
  • motivation, history

• Foundations of DPD
  • Coarse-graining of MD (Mori-Zwanzig formalism)

• Parameterization of a DPD system
  • Force field of classic DPD
  • Pressure, compressibility, viscosity, diffusivity of DPD fluids
  • Scaling
  • Modeling of complex fluids using DPD

• How do you practically run a DPD simulation
  • Implementation of boundary conditions
  • Computing viscosity, diffusivity

• Some Applications
  • Droplet, Blood flows, Self-assembly Dynamics
Parameterization of a DPD system

**Force field of classic DPD**

\[
F_i = \sum_{i \neq j} \left( F^C_{ij} + F^D_{ij} + F^R_{ij} \right)
\]

\[
F^C_{ij} = a\left(1 - \frac{r_{ij}}{r_c}\right)e_{ij}
\]

\[
F^D_{ij} = \gamma\left(1 - \frac{r_{ij}}{r_c}\right)^2 (e_{ij}v_{ij})e_{ij}
\]

\[
F^R_{ij} = \sqrt{2\gamma k_B T \left(1 - \frac{r_{ij}}{r_c}\right)} dt^{-1/2} \xi_{ij} e_{ij}
\]

The conservative force \( F^C_{ij} \) is responsible for the static properties, i.e.,

- Pressure
- Compressibility
- Radial distribution function \( g(r) \)

The dissipative force \( F^D_{ij} \) and random force \( F^R_{ij} \) together act as a thermostat and determine the dynamics properties, i.e.,

- Viscosity
- Diffusivity
- Time correlation functions
Parameterization of a DPD system

DPD thermostat

\[ F_i = \sum_{i \neq j} (F^C_{ij} + F^D_{ij} + F^R_{ij}) \]

\[ F^D_{ij} = \gamma w_D(r_{ij})(e_{ij}v_{ij})e_{ij} \]

\[ F^R_{ij} = \sigma w_R(r_{ij})dt^{-1/2} \xi_{ij} e_{ij} \]

To satisfy the fluctuation-dissipation theorem (FDT):

\[ [w_R(r)]^2 = w_D(r) \quad \text{and} \quad \sigma^2 = 2\gamma k_B T \]

Then, the dissipative force \( F^D_{ij} \) and random force \( F^R_{ij} \) together act as a DPD thermostat.

![Graph showing time step vs. \( k_B T \)]
RDF, pressure, compressibility, viscosity, diffusivity of DPD fluids

Radial distribution function

\[ g(r) = \frac{n(r)}{4\pi r^2 \Delta r \rho} \]

Pressure

\[ P = \rho k_B T + \frac{1}{3V} \left( \sum_i r_i \cdot F_i \right) \]
\[ P = \rho k_B T + \frac{2\pi \rho^2}{3} \int_0^\infty r^3 f(r)g(r) \, dr \]

Compressibility

\[ \kappa^{-1} = \frac{1}{k_B T} \left( \frac{\partial P}{\partial \rho} \right)_T \]

For linear conservative force

\[ \mathbf{F}^C_{ij} = a(1 - r_{ij}/r_c) \mathbf{e}_{ij} \]

The equation of state is

\[ P = \rho k_B T + 0.1a\rho^2 \]

Then

\[ \kappa^{-1} = 1 + 0.2a\rho/k_B T \]
**RDF, pressure, compressibility, viscosity, diffusivity of DPD fluids**

**Diffusivity**

Consider the motion of single particle given by Langevin equation

\[ m \frac{dv}{dt} = -\frac{v_i}{\tau} + F_i^R \]

\[ \frac{1}{\tau} = \sum_{j \neq i} \gamma w_D(r) \frac{e_{ij} e_{ij}}{3} = \frac{4\pi \gamma \rho}{3} \int_0^\infty r^2 w_D(r) g(r) dr \]

Self-diffusion coefficient

\[ D = \frac{1}{3} \int_0^\infty \langle v(t) v(0) \rangle dt = \tau k_B T \]

**Viscosity**

There are two contributions to the pressure tensor: the *kinetic* part \( \nu_K \) and the *dissipative* part \( \nu_D \)

\[ \nu_K = \frac{D}{2} \]

\[ \nu_D = \frac{2\pi \gamma \rho}{15} \int_0^\infty r^4 w_D(r) g(r) dr \]

If \( w_D(r) = (1 - r/r_c)^2 \), and using \( g(r) = 1 \), we have

\[ \nu = \frac{45k_B T}{4 \pi \gamma \rho r_c^3} + \frac{2\pi \gamma \rho r_c^5}{1575} \]


Scaling

Basic units: Length, mass and time units

- The mass of the DPD particle is $N_m$ times the mass of MD particle.
  \[ M = mN_m \]

- The cut-off radius can be found by equating mass densities of MD and DPD systems.
  \[ R_c = \left( \frac{N_m \rho_{DPD}^*}{\rho_{MD}} \right)^{\frac{1}{3}} \sigma \]

- The DPD conservative force coefficient $a$ is found by equating the dimensionless compressibility of the systems.
  \[ a = k_B T \frac{1}{2} \frac{\alpha_{DPD}^{-1} N_{D}}{\rho_{MD}} \]

- The time scale is determined by insisting that the shear viscosities of the DPD and MD fluids are the same.
  \[ \tau_{DPD} = \frac{\nu_{DPD}^*}{\nu_{MD}} \left( \frac{R_c}{\sigma} \right)^2 \tau \]

- The variables marked with the symbol "*" have the same numerical values as in DPD but they have units of MD.

Modeling of complex fluids using DPD

**Bonded interactions**

**Hookean spring** (harmonic potential)

\[ V = K_b (r - r_0)^2 \]

\[ F = - \frac{dV}{dr} = 2K_b(r_0 - r) \]

**Finitely extensible nonlinear elastic (FENE) spring**

\[ V = -0.5KR_0^2 \ln \left[ 1 - \left( \frac{r}{R_0} \right)^2 \right] + 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right] + \epsilon \]

**Wormlike chain (WLC)** (for semi-flexible polymers)

An interpolation formula that approximates the force-extension behavior is (Marko-Siggia formula)

\[ F(r) = \frac{k_BT}{l_p} \left[ \frac{1}{4} \left( 1 - \frac{r}{L} \right)^{-2} + \frac{r}{L} - \frac{1}{4} \right] \]

*L* represents the contour length of the polymer chain. *l_p* is persistence length, indicating the rigidity/flexibility of the polymer chain.

Part II: Dissipative Particle Dynamics

- Brief introduction of DPD
  - motivation, history
- Foundations of DPD
  - Coarse-graining of MD (Mori-Zwanzig formalism)
- Parameterization of a DPD system
  - Force field of classic DPD
  - Compressibility, viscosity, diffusivity of DPD fluids
  - Scaling
  - Modeling of complex fluids using DPD

- How do you practically run a DPD simulation
  - Implementation of boundary conditions
  - Computing viscosity, diffusivity

- Some Applications
  - Droplet, Blood flows, Self-assembly Dynamics
Modified velocity-Verlet algorithm for DPD

DPD forces depend on velocity

\[ r_i(t + \Delta t) = r_i(t) + \Delta t v_i(t) + \frac{1}{2} (\Delta t)^2 f_i(t) \]

\[ \tilde{v}_i(t + \Delta t) = v_i(t) + \lambda \Delta t f_i(t) \]

\[ f_i(t + \Delta t) = f_i(r(t + \Delta t), \tilde{v}(t + \Delta t)) \]

\[ v_i(t + \Delta t) = v_i(t) + \frac{1}{2} \Delta t (f_i(t) + f_i(t + \Delta t)) \]

Optimum value:

\[ \lambda = 0.65 \]

For this value the time step can be increased to 0.06 without significant loss of temperature control.

Implementation of Boundary conditions: PBC

**Periodic boundary conditions**

For computational box $0 < x < L$

If $x < 0$ then $x = x + L$
If $x > L$ then $x = x - L$

**Minimum image criterion**: among all images of a particle, consider only the closest and neglect the rest.

**Lees-Edwards boundary conditions**

$\gamma L = \frac{d\gamma_z}{dy}$

Is replaced by one here, shifted over toward the edge of the cell

Shift distance = $\gamma L t$
Implementation of Boundary conditions: wall

Solid objects in DPD are made of frozen particles

Soft repulsion in DPD cannot prevent particles from penetrating walls

Specular Reflection

Maxwellian Reflection

Bounce-back Reflection

As coarse-graining parameter $N_m$ increases, the density fluctuations in the DPD simulations also increase, with the $N_m=5$ case exhibiting very large values at the wall and also inside the flow domain.

This is not a desired effect, because we expect the fluctuations to decrease as we approach the continuum, i.e. $N_m \to \infty$.

Iteratively adjust the wall repulsion force in each bin based on the averaged density values.

Adaptive BC:
- layers of particles
- bounce back reflection
- adaptive wall force

Implementation of Boundary conditions: wall

**Effective boundary forces**

\[ F_C(h) = 2\pi \rho n_w \int_{z=h}^{r_c} \int_{x=0}^{\sqrt{r_c^2-z^2}} F^C(r)g(r)z \cdot dx \cdot dz = f^C(h)n_w \]

\[ F_D\parallel(h) = -\pi \rho yue_x \int_{z=h}^{r_c} \int_{x=0}^{\sqrt{r_c^2-z^2}} \omega_D(r)g(r)x^3 \frac{z}{hr^2} \cdot dx \cdot dz \]

\[ F_D\perp(h) = -2\pi \rho ywe_z \int_{z=h}^{r_c} \int_{x=0}^{\sqrt{r_c^2-z^2}} \omega_D(r)g(r)xz^2 \frac{z}{hr^2} \cdot dx \cdot dz \]

\[ F_R(h) = (\sigma_{R\parallel}(h)e_x + \sigma_{R\perp}(h)e_z) \xi \]

Time-evolution of the velocity profile in Poiseuille flow
Computing viscosity, diffusivity

**Viscosity**

**Plane Poiseuille flow**
\[
\frac{du}{dt} = -\frac{1}{\rho} \frac{dp}{dx} + g_x + \nu \frac{d^2u}{dz^2}
\]
Steady case and driven by \(g_x\):
\[
-\nu \frac{d^2u}{dz^2} = g_x
\]

**Periodic Poiseuille flow**
\[
-\nu \frac{d^2u}{dz^2} = g_x
\]
BC: \(u(z = 0) = 0\)
\(u(z = \pm d) = 0\)
Solution:
\[
\frac{g_x}{2\nu} z(d - |z|)
\]

**Couette flow (constant shear rate)**
\[
\tau_{yx} = \mu \frac{du}{dy} = \mu \dot{\gamma}
\]

**Diffusivity**
\[
D = \lim_{t \to \infty} \frac{1}{6t} \left( |\mathbf{r}(t) - \mathbf{r}(0)|^2 \right)
\]
\[
D = \frac{1}{3} \int_0^\infty \langle \mathbf{v}(t) \mathbf{v}(0) \rangle dt
\]
Part II: Dissipative Particle Dynamics

- Brief introduction of DPD
  - motivation, history
- Foundations of DPD
  - Coarse-graining of MD (Mori-Zwanzig formalism)
- Parameterization of a DPD system
  - Force field of classic DPD
  - Pressure, compressibility, viscosity, diffusivity of DPD fluids
  - Scaling
  - Modeling of complex fluids using DPD

- How do you practically run a DPD simulation
  - Implementation of boundary conditions
  - Computing viscosity, diffusivity

- Some Applications
  - Droplet, Blood flows, Self-assembly Dynamics
Some Applications: Droplet


From the website of LSST of ETH Zürich

Some Applications: Blood flows

http://www.dam.brown.edu/people/ytang/index.php

Provided by Xuejin Li and Lu Lu of CRUNCH Group
Some Applications: Self-assembly Dynamics

$10$ particles

$15000$ particles

$124000$ particles

Some Applications: Self-assembly Dynamics

$128M$ DPD particles

$100$ particles

$10000$ particles

$124000$ particles

GPU simulations performed by Yu-Hang Tang of CRUNCH Group
Software/Package for DPD simulation

**LAMMPS**: Large-scale Atomic/Molecular Massively Parallel Simulator
http://lammps.sandia.gov/

**ESPResSo**: Extensible Simulation Package for Research on Soft matter http://espressomd.org/

**HOOMD-blue**: a general-purpose particle simulation toolkit
http://codeblue.umich.edu/hoomd-blue/

**DPDmacs**: Fast coarse-grained simulations
http://www.apmaths.uwo.ca/~mkarttu/dpdmacs.shtml
Features: compatible with Gromacs.

**MyDPD**: C++ mesodynamics code
http://multiscalelab.org/mydpd
Features: simple, serial but functional.